

A) Implementar un simulador que determine la medida de rendimiento para el entorno del mundo de la aspiradora según las siguientes especificaciones:

1. La medida de rendimiento premia con un punto al agente por cada recuadro que limpia (aspira) en un período de tiempo concreto, a lo largo de una «vida» de 1.000 períodos.
2. La «dimensión» de la grilla se conoce a priori pero la distribución de la suciedad y la localización inicial del agente no se conocen (aleatorio). Las cuadrículas se mantienen limpias y aspirando se limpia la cuadrícula en que se encuentra el agente
3. Las acciones Izquierda, Derecha, Arriba, Abajo mueven al agente en dichas direcciones, excepto en el caso en que lo pueda llevar fuera de la grilla.
4. Las acciones permitidas son:
 - a. Arriba
 - b. Abajo
 - c. Izquierda
 - d. Derecha
 - e. Limpiar
 - f. NoHacerNada
5. El agente percibe su locación y si esta contiene suciedad

Posibles interfaz a utilizar

```
class Environment:
    def __init__(self, sizeX, sizeY, init_posX, init_posY, dirt_rate)

    def accept_action(self, action):
    def is_dirty(self):
    def get_performance(self):
    def print_environment(self):
```

B) Implementar un agente reflexivo simple para el entorno de la aspiradora del ejercicio anterior.

Posible interfaz para el Agente

```
class Agent:
    def __init__(self, env): #recibe como parámetro un objeto de la clase
Environment
    def up(self):
    def down(self):
```

```
def left(self):  
def right(self):  
def suck(self): # Limpia  
def idle(self): # no hace nada  
def prespective(self,env): #sensa el entorno  
def think(self): # implementa las acciones a seguir por el agente
```

C) Evaluar el desempeño del agente agente reflexivo para:

1. Entornos de : 2x2, 4x4, 8x8, 16x16, 32x32, 64x64, 128x128
2. Porcentaje de Suciedad en el ambiente: 0.1, 0,2 0,4, 0.8

D) Repetir el procedimiento descrito en el punto C, para el caso de un agente con comportamiento totalmente aleatorio.

E) Responder preguntas 2.10 y 2.11 de AIMA.